[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR DIGITAL IMAGE SEGMENTATION
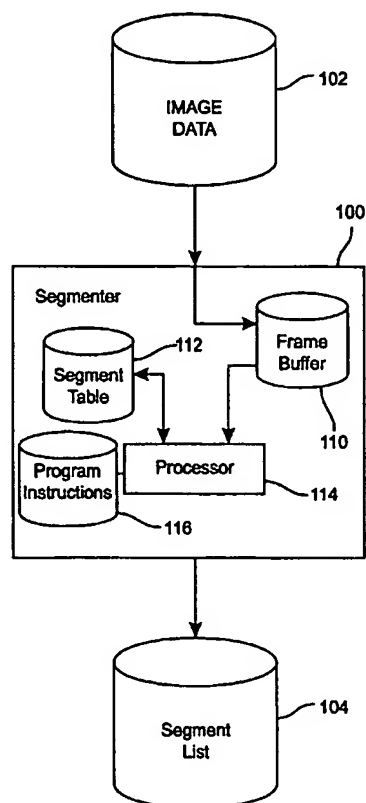
(57) Abstract: An image segmenter uses one or more techniques to accurately segment an image, including the use of a progressive flood fill to fill incompletely bounded segments, the use of a plurality of scaled transformations and guiding segmentation at one scale with segmentation results from another scale, detecting edges using a composite image that is a composite of multiple color planes, generating edge chains using multiple classes of edge pixels, generating edge chains using the plurality of scaled transformations, and/or filtering spurious edges at one scale based on edges detected at another scale.

BEST AVAILABLE COPY

WO 00/77735 A1

patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European
patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,
IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG,
CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*

— *Before the expiration of the time limit for amending the
claims and to be republished in the event of receipt of
amendments.*

*For two-letter codes and other abbreviations, refer to the "Guid-
ance Notes on Codes and Abbreviations" appearing at the begin-
ning of each regular issue of the PCT Gazette.*

METHOD AND APPARATUS FOR DIGITAL IMAGE
SEGMENTATION

5

CROSS-REFERENCES TO RELATED APPLICATIONS

This application claims priority from U.S. Provisional Patent Application No. 60/139,134, filed June 11, 1999, the disclosure of which is incorporated herein in its entirety by reference for all purposes.

10

FIELD OF THE INVENTION

The present invention relates to image processing in general, and more particularly to the problem of image segmentation where an image needs to be automatically segmented into segments based on the pixel color values of the image.

BACKGROUND OF THE INVENTION

15

Image segmentation is the process of partitioning an image into a set of non-overlapping parts, or segments, that together constitute the entire image. Image segmentation is useful for many applications, one of which is machine learning.

In machine learning, an image is segmented into a set of segments and a designated segment from the image or another image is compared with the set of

20

segments. When a machine successfully matches the designated segment with one or more segments from a segmented image, the machine draws an appropriate conclusion. For example, image segmentation could be used to identify misshapen blood corpuscles for determination of blood diseases such as sickle cell anemia. In this example, the designated segment would be a diseased blood cell. By counting the number of segment

25

matches in a given image, the relative health of a patient's blood can be determined. Other applications include compression and processes that process areas of the image in ways that depend on the areas' segments.

As the terms are used herein, an image is data derived from a multi-dimensional signal. The signal might be originated or generated either naturally or artificially. This

30

multi-dimensional signal (where the dimension could be one, two, three, or more) may be represented as an array of pixel color values such that pixels placed in an array and colored according to each pixel's color value would represent the image. Each pixel has a

1

location and can be thought of as being a point at that location or as a shape that fills the area around the pixel such that any point within the image is considered to be "in" a pixel's area or considered to be part of the pixel. The image itself might be a multidimensional pixel array on a display, on a printed page, an array stored in memory,

5   or a data signal being transmitted and representing the image. The multidimensional pixel array can be a two-dimensional array for a two-dimensional image, a three-dimensional array for a three-dimensional image, or some other number of dimensions.

       The image can be an image of a physical space or plane or an image of a

10   simulated and/or computer-generated space or plane. In the computer graphic arts, a common image is a two-dimensional view of a computer-generated three-dimensional space (such as a geometric model of objects and light sources in a three-space). An image can be a single image or one of a plurality of images that, when arranged in a suitable time order, form a moving image, herein referred to as a video sequence.

15   When an image is segmented, the image is represented by a plurality of segments. The degenerate case of a single segment comprising the entire image is within the definition of segment used here, but the typical segmentation divides an image into at least two segments. In many images, the segmentation divides the image into a background segment and one or more foreground segments.

20   In one segmentation method, an image is segmented such that each segment represents a region of the image where the pixel color values are more or less uniform within the segment, but dramatically change at the edges of the image. In that implementation, the regions are connected, i.e., it is possible to move pixel-by-pixel from any one pixel in the region to any other pixel in the region without going outside the

25   region.

       Pixel color values can be selected from any number of pixel color spaces. One color space in common use is known as the YUV color space, wherein a pixel color value is described by the triple (Y, U, V), where the Y component refers to a grayscale intensity or luminance, and U and V refer to two chrominance components. The YUV color space

30   is commonly seen in television applications. Another common color space is referred to as the RGB color space, wherein R, G and B refer to the Red, Green and Blue color components, respectively. The RGB color space is commonly seen in computer graphics representations, along with CYMB (cyan, yellow, magenta, black) often used with computer printers.

An example of image segmentation is illustrated in Fig. 1. There, an image 10 is of a shirt 20 on a background 15. The image can be segmented into segments based on colors (the shading of shirt 20 in Fig. 1 represents a color distinct from the colors of background 15 or pockets 70, 80). Thus, background 15, shirt 20, buttons 30, 40, 50, 60

5   and pockets 70, 80 are segmented into separate segments in this example. In this example, if each segment has a very distinct color and the objects in image 10 end cleanly at pixel boundaries, segmentation is a simple process. In general, however, generating accurate image segments is a difficult problem and there is much open research on this problem, such as in the field of "computer vision" research. One reason segmentation is

10 often difficult is that a typical image includes noise introduced from various sources including, but not limited to, the digitization process when the image is captured by physical devices and the image also includes regions that do not have well-defined boundaries.

There are several ways of approaching the task of image segmentation, which can

15 generally be grouped into the following: 1) histogram-based segmentation; 2) traditional edge-based segmentation; 3) region-based segmentation; and 4) hybrid segmentation, in which several of the other approaches are combined. Each of these approaches is described below.

1. Histogram-based Segmentation

20 Segmentation based upon a histogram technique relies on the determination of the color distribution in each segment. This technique uses only one color plane of the image, typically an intensity color plane (also referred to as the greyscale portion of the image), for segmentation. To perform the technique a processor creates a histogram of the pixel color values in that plane. A histogram is a graph with a series of "intervals"

25 each representing a range of values arrayed along one axis and the total number of occurrences of the values within each range shown along the other axis. The histogram can be used to determine the number of pixels in each segment, by assuming that the color distribution within each segment will be roughly a Gaussian, or bell-shaped, distribution and the color distribution for the entire image will be a sum of Gaussian

30 distributions. Histogram-based techniques attempt to recover the individual Gaussian curves by varying the size of the intervals, i.e., increasing or decreasing the value range, and looking for high or low points. Once the distributions have been ascertained, then each pixel is assigned to the segment with its corresponding intensity range.

The histogram method is fraught with errors. The fundamental assumption that the color distribution is Gaussian is at best a guess, which may not be accurate for all images. In addition, two separate regions of identical intensity will be considered the same segment. Further, the Gaussian distributions recovered by the histogram are

5   incomplete in that they cut off at the ends, thus eliminating some pixels. Further, this method of segmentation is only semi-automatic, in that the technique requires that the number of segments are previously known and that all of the segments are all roughly the same size.

         2. Traditional Edge-Based Segmentation

10   Traditional edge-based segmentation uses differences in color or greyscale intensities to determine edge pixels that delineate various regions within an image. This approach typically assumes that when edge pixels are identified, the edge pixels will completely enclose distinct regions within the image, thereby indicating the segments. However, traditional edge detection techniques often fail to identify all the pixels that are

15   in fact edge pixels, due to noise in images or other artifacts. If some edge pixels are missed, some plurality of distinct regions might be misidentified as being a single segment.

         3. Region-based Segmentation

         Region based segmentation attempts to detect homogenous regions and designate

20   them as segments. One class of region-based approaches starts with small uniform regions within the image and tries to merge neighboring regions that are of very close color value in order to form larger regions. Conversely, another class of region-based approaches starts with the entire image and attempts to split the image into multiple homogeneous regions. Both of these approaches result in the image being split at regions

25   where some homogeneity requirements are not met.

         The first class of region based segmentation approaches is limited in that the segment edges are approximated depending on the method of dividing the original image. A problem with the second class of region based approaches is that the segments created tend to be distorted relative to the actual underlying segments.

30   4. Hybrid Segmentation

         The goal of hybrid techniques is to combine processes from multiple previous segmentation processes to improve image segmentation. Most hybrid techniques are a combination of edge segmentation and region-based segmentation, with the image being segmented using one of the processes and being continued with the other process. The

hybrid techniques attempt to generate better segmentation than a single process alone. However, hybrid methods have proven to require significant user guidance and prior knowledge of the image to be segmented, thus making then unsuitable for applications requiring fully automated segmentation.

5   SUMMARY OF THE INVENTION

The present invention solves many of the problems of previous segmentation processes. In an image segmenter according to one embodiment of the present invention, the image segmenter uses one or more techniques to accurately segment an image, including the use of a progressive flood fill to fill incompletely bounded segments, the

10   use of a plurality of scaled transformations and guiding segmentation at one scale with segmentation results from another scale, detecting edges using a composite image that is a composite of multiple color planes, generating edge chains using multiple classes of edge pixels, generating edge chains using the plurality of scaled transformations, and/or filtering spurious edges at one scale based on edges detected at another scale.

15   A further understanding of the nature and the advantages of the inventions disclosed herein may be realized by reference to the remaining portions of the specification and the attached drawings.

                            BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is an image illustrating a simple image segmentation process.

20   Fig. 2 is a block diagram of an apparatus for segmenting images.

Fig. 3 is a block diagram of a system in which a segmented image might be used.

Fig. 4 is an illustration of a data stream comprising an image and related segment

data.

Fig. 5 illustrates how a segment list might appear for a corresponding image.

25   Fig. 6 illustrates edge pixels within an image of black and white image pixels. The edge pixels, illustrated by the smaller, lightly shaded pixels, lie between the black and white image pixels.

Fig. 7 illustrates an edge chain between two segments of image pixels.

Fig. 8 shows an image to be segmented using a progressive brush.

30   Fig. 9 is an illustration of an image that is partially filled on either side of an edge chain.

Fig. 10 illustrates gradients for pixels in each of the three color components.

Fig. 11 shows an image with image pixels, edge pixels and an edge.

Fig. 12 illustrates a process of determining strong edge pixels using a gradient technique.

Fig. 13 illustrates a process of determining weak edge pixels using a gradient technique.

Fig. 14 illustrates a process of determining strong edge pixels using a Laplacian technique.

Fig. 15 shows an image with image pixels and edge pixels identified, including strong edge pixels and weak edge pixels.

Fig. 16 illustrates a process for selecting among edge pixels in generating an edge chain.

Fig. 17 illustrates a process for continuing edge chains over small gaps; Fig. 17(a) shows two edge chains with a gap; Fig. 17(b) shows the gap filled in.

Fig. 18 illustrates a process for linking edge chains from more than one scale; Fig. 18(a) shows an edge chain from a coarser scale; Fig. 18(b) shows an edge chain from a finer scale; and Fig. 18(c) shows their combination.

Fig. 19 illustrates a process of edge chain extension; Fig. 19(a) shows an image before edge chains are extended; Fig. 19(b) shows the image after edge chains are extended.

Fig. 20(a)-(e) are images with various degrees of edge chain filtering.

Figs. 21(a)-(c) illustrate a process of edge chain filtering over video frames.

DESCRIPTION OF THE SPECIFIC EMBODIMENTS

Segmentation is the process by which a digital image is subdivided into components referred to as "segments" of the image. In a color value based segmentation process, each segment represents an area bounded by radical or sharp changes in color values within the image, as shown in Fig. 1 and as described within the background. In many examples described in detail herein, the image represents a two-dimensional signal, but it should be understood that the methods and apparatus described herein can be adapted for other numbers of dimensions by one of skill in the art after reading this disclosure.

Fig. 2 is a block diagram of a system including a segmenter 100 that generates segment definitions for an image according to one embodiment of the present invention. Segmenter 100 accepts as its input image data 102 and outputs a segment list 104. The

format of image data 102 and segment list 104 can vary depending on the nature of the image, its storage requirements and other processing not related the segmentation process, but one form of storage for image data 102 is as an array of pixel color values, possibly compressed, and stored in one of many possible industry-standard image formats, such as

5   raw data, bitmaps, MPEG, JPEG, GIF, etc. In memory, image data 102 might be stored as a two-dimensional array of values, where each value is a pixel color value. The pixel color value might have several components. For example, an image might be a 1024 by 768 array of pixels, with each pixel's color value represented by three (red, green, blue) component values ranging from 0 to 255. The format of segment list 104 might be stored

10   as a run-length encoded ordered list of midpixels (defined below with reference to Fig. 6) or image pixels that comprise the bounds of each segment.

Segmenter 100 is shown comprising a frame buffer 110 that holds the image data as it is being considered, a segment table 112 that holds data about the segments identified or to be identified, and a processor 114 that operates on frame buffer 110 to

15   generate segment data according to program instructions 116 provided in segmenter 100. Several aspects of program instructions 116 are described below and might include program instructions corresponding to some or all of the methods and processes for segmentation and in support of segmentation described herein.

Fig. 3 illustrates a system in which segment list 104 might be used. As shown

20   there, an image generator 200 generates an image, possibly using conventional image generation or image capture techniques, and stores data representing that image as image data 102. A segmenter, such as segmenter 100 shown in Fig. 2, is used to generate segment list 104 as described above. Image generator 200 provides segment list 104 to a segment field generator 201 that generates data for each of the segments. Such data

25   might include a label, a clickable link (such as a Uniform Resource Locator, or "URL"), and other data not necessarily extracted from the image but associated with segments of the image.

Image data 102, segment list 104 and the segment fields are stored as web pages to be served by a web server 202. That image and related data can then be retrieved from

30   web server 202 over Internet 204 by a browser 206 or other web client (not shown).

Referring now to Fig. 4, one arrangement of image data and the related data as might be transmitted as a data signal are shown in Fig. 4. In this example, the image data 250 is transmitted as a signal (possibly in an industry-standard format) followed by the segment list 260 and segment fields 270.

Fig. 5 illustrates an extremely simple image and its resulting segmentation. As shown, image 500 contains three segments, two polygonal foreground segments and a background segment. Segmenter 100 processes image 500 to generate segment list 510. As shown in the figure, the segment labelled "Seg #1" represents the background segment, the segment labelled "Seg #2" represents the foreground segment bounded by a polygon between points A, B, C, E and D (ending with point A to close the polygon), and the segment labelled "Seg #3" represents the foreground segment bounded by a polygon between points F, G, K, J, I and H (ending with point F to close the polygon). Of course, the typical image being segmented is not usually so well defined, so some or all of the methods described herein might be needed to correctly identify segment boundaries.

As used herein, the term "midpixel" refers to a logical point located in an image relative to image pixels. An edge of a segment runs from midpixel to midpixel, thus separating image pixels on each side of the segment. Midpixels preferably do not lie on the same points on which image pixels lie, but fall between image pixels. While it is not required that midpixels be exactly centered in a rectangle defined by four mutually adjacent image pixels (or other minimum polygon defined by mutually adjacent image pixels on nonrectangular image pixel arrays), without loss of generality, centered midpixels are preferred for the simplicity of arrangement.

Such an arrangement is shown in Fig. 6, where an image 600 comprises image pixels, such as image pixels 602, and midpixels 604 occur between image pixels. As should be apparent, if an edge were specified connecting all midpixels 604 in an order running from top to bottom, or bottom to top, the edge would separate image pixels on the left of the edge from image pixels on the right of the edge.

Where a midpixel is part of an edge, or proposed as part of an edge, it is referred to as an edge pixel. Thus, edges are chains of edge pixels and the image pixels of a segment can be bounded by an edge chain surrounding those image pixels. In some cases, a segment includes image pixels that are exactly on an edge. When an image pixel falls exactly on an edge between two segments (as might occur with a diagonal edge and centered edge pixels), one or more of various tie-breaking routines can be employed to determine to which segment the image pixel should belong. As explained more fully herein, edge pixels are the approximate points in an image where the pixel color values undergo relatively large shifts.

8

The segmentation process described below comprises several sequences of steps. Not all of the steps need be performed and those that are performed need not be performed in the order listed. The sequences that are initially described are:

1) Progressive Flood Fill - using varying brush sizes to progressively fill regions that might be incompletely bounded;

2) Multiscale Segmenting - creating a plurality of "scale" transformations of the original image and using segments from one scale transformation to guide segmenting at another scale;

3) Composite Edge Detection - detecting edges using a composite image that is a composite of multiple color planes;

4) Multi-class Edge Chaining - generating edge chains using multiple classes of edge pixels;

5) Multiscale Edge Chaining - using information from multiple scales to generate edge chains;

6) Edge Chain Filtering - using various contextual edge characteristics such as multi-scaling, video sequencing, dynamic scales, to filter extraneous edge chains.

The above steps will now be described in more detail, with reference to the figures as needed, followed by some particularly useful combinations of the above steps.

1. Progressive Flood Fill

The progressive flood fill process generates closed segment bounds from possibly incomplete edge chains. This process assumes an image with at least some edge chains, where an edge chain is an ordered list of edge pixels logically connected with line segments between edge pixels adjacent in the ordered list. The edge chains for a given image can be generated in a number of ways, including an edge chain generation process described herein. Fig. 7 shows an example of an edge chain 650. Edge chain 650 is defined by the six edge pixels 652 and the line segments that connect the edge pixels.

In general, the progressive flood fill process described immediately below uses a sequence of brushes, from large to small, to "fill" prospective segments. A brush is a logical window or a given shape expressed in pixel units. For example, one possible brush is six-by-six pixel square window, another is a hexagon window, with seven pixels per side. Filling a prospective segment is a process of covering image pixels with the brush, assigning the image pixels a segment value (i.e., a value or number that can be used as a reference to a segment) and moving the brush around the image without the

brush covering any portion of an edge chain and without the brush covering a differentially assigned image pixel, i.e., an image pixel previously covered and assigned a segment value different from the segment value currently being assigned to covered image pixels. The set of image pixels that can be reached by the sequence of brushes

5   without covering an edge chain or a differentially assigned image pixel is a set of image pixels associated with the prospective segment.

For example, consider Fig. 8, which shows an image 700 and a brush 702, where the image has edge chains 701 and 701a as marked. Given the size of brush 702, if brush 702 is placed in the triangular area 704, then the filling process will not "bleed" through

10   the gap 705 at the top of area 704, because brush 702 cannot fit through gap 705 without covering part of an edge chain. The first step of the progressive fill process brushes over area 704, square area 706 and background area 708, using a brush of the size indicated. One result of that first step is to associate some, in this case most, of the pixels of the image with respective segments. Some of the image pixels are not associated with

15   segments because brush 702 could not reach those pixels without covering an edge chain.

In one implementation, the processor moves the brush to each accessible location within the image, taking care not to cover an edge chain or a differentially assigned image pixel. At each brush location, the processor considers the underlying midpixels and image pixels. If the brush covers an edge pixel or an edge chain, or a differentially

20   assigned image pixel, the brush has no effect and is moved to the next position. However, if the brush does not cover any portion of an edge pixel, edge chain or a differentially assigned image pixel, the processor examines the underlying image pixels. If any one of the underlying image pixels has already been associated with a segment, all the underlying image pixels are assigned to that segment, otherwise all the underlying

25   image pixels are assigned to a new segment. In the case of a one pixel brush, the processor considers the adjacent image pixels and assigns the image pixel to the segment having the most adjacent image pixels, without crossing any edge chains.

In another implementation, the processor moves two brushes on either side of each detected edge chain, assigning image pixels on either side of the edge chains to

30   different segments. In yet another implementation, the processor places a brush at any unprocessed location on the image and moves the brush to adjacent locations that can be reached without covering an edge chain.

In another implementation, the processor uses a sequence of incrementally smaller brushes to alternatively create new segments and expand previous segments. Thus, odd

numbered passes might create new segments while even numbered passes expand previously created segments.

In yet another implementation, the processor creates a sequence of edge chains in each color component and combines the edge chains to create a composite edge chain

5   picture. The combination may either be the union or intersection of the edge chains in each color component.

Once the processor passes the first brush (or brushes) over the image, some of the image pixels are assigned to segments. These image pixels represent the portions of the image that were accessible to the brush(es). The processor then makes another pass over

10   the image, using a smaller brush. In this second pass, the processor performs the same process with the smaller brush, to cover and assign image pixels that were not "reachable" by the larger brush (i.e., the brush could not cover the image pixels without also covering an edge pixel, edge chain or a differentially assigned image pixel). The process is repeated in subsequent passes, with increasingly smaller brushes until the processor

15   makes a pass with the smallest brush, such as a one-pixel brush.

Preferably, the initial brush size is at least one pixel larger than the largest acceptable gap in an edge chain defining a segment. For example, if a span of five pixels or less between the ends of two edge chains is considered a gap in one larger edge chain, then the initial brush might be a six-by-six pixel square. With such an arrangement, the

20   initial brush would not "bleed" through a gap to incorrectly combine two segments. After the initial brush, subsequent, smaller brushes might be able to bleed through a gap, but the amount of bleeding through would be limited, because on both sides of the gap, there would be previously assigned image pixels which would limit the brush's movements.

Fig. 9 illustrates this point. That figure shows an edge chain 800 with a gap.

25   Edge chain 800 separates two segments 802, 804. In an earlier brush pass, some of the image pixels were assigned to segments and some image pixels, such as those near the gap, were not reachable by the brush. When a smaller brush is used, the brush might be small enough to reach all the remaining unassigned pixels on both sides of the gap, resulting in a bleed through of whichever segment is processed first. To prevent this, the

30   processor will pass two brushes over the unassigned pixels, from either side of the gap, so that both sides of the gap are filled evenly.

To further limit undesirable bleeding of smaller brushes, two brushes can be run in a pass, one on each side of the gap, or one brush can alternate from side to side in the process of assigning image pixels to segments. In some cases, such as for image pixel

11

660 shown in Fig. 7, even a one-pixel brush is not small enough to reach some image pixels, such as those that are crossed by an edge chain. Those unreachable pixels can be dealt with after the pass using the smallest brush is complete.

One process for dealing with unreachable pixels is to use a tie-breaking scheme to

5   assign an unreachable pixel to one of the segments that meet at the unreachable pixel. One tie breaking scheme considers the gradient at the image pixel and the gradient at the closest image pixel in each of the contending segments and the image pixel is assigned to the segment that contains the closest image pixel with the gradient closest in magnitude and direction to the gradient at the unreachable image pixel. A gradient is a vector

10   derivative.

Once the process associates each image pixel with a segment, the locations of the boundaries of each segment are easily found. As described below, the progressive fill process might be combined with other processes to more accurately determine segment boundaries.

15   2. Multiscale Segmenting

Starting with an image, the segmenter generates a plurality of transformations of the image at progressively lower resolutions, also known as "scales", keeping only information regarding the larger, more dominant features from scale to scale. There are several different ways to generate the transformations, such as the use of smoothing or

20   similar filters. One set of transformations that might be useful for some images is an array of Gaussian smoothing filters, where each filter has a different characteristic distance.

In one multiscale segmenting process, the transformed images are processed in order from coarsest to finest, where the coarsest image is the transformation using the

25   widest smoothing filter. Typically, the coarsest image only retains the larger features of the image. The finest image is either the image transformed with the smoothing filter with the smallest characteristic distance or the original, untransformed image.

The process continues by running a segmentation process of the coarsest image to define a set of segments for the coarsest image. The segmentation process can be the

30   progressive fill segmentation process described above, or some other segmentation process performed on single images. Once the segmentation process is performed on the coarsest image, that image's set of segments is used in subsequent segmentation processes performed on finer images.

12

For example, the second coarsest image is segmented as before, but with a constraint that a segment in the second coarsest image cannot encompass more than one segment in the coarsest image. In other words, the segments in the second image are each subset of the segments in the first image.

5   The process continues for each next coarser image, using the segments of the previously segmented image. Since the segments of each prior image were constrained to be subsets of its prior image, the segments of any one of the transformed images are subsets of the segments of any segment in any coarser image.

One method of enforcing the "subset" constraint is to perform an unrestricted

10   segmentation of an image, then subdivide any segment that crosses more than one segment in a coarser image. Another method of enforcing the subset constraint might be used where segmentation is done by the progressive fill process described above. In this latter method, the segment boundaries of the coarser image are added as edge chains in the image being processed, to effect the subset constraint.

15   3. Composite Edge Detection

The above-described progressive fill process and multiscale segmentation process operate on an image and a set of edge chains, where an edge chain is an ordered set of edge pixels. The edge chains are generated from the edge pixels. The above-described methods might use other methods of edge detection, but one method that is particularly

20   useful when pixel color values comprise multiple color components is the composite edge detection process that will now be described.

In the composite edge detection process, information from the different color components is combined to form a composite image, where each pixel color value in the composite image is a function of the components of the color values of the corresponding

25   image pixel and possibly the color values of surrounding image pixels. The composite image is then used to determine which of the midpixels are edge pixels. Once the edge pixels are determined in the composite image, the edge pixels can be linked into edge chains.

Three methods of composite image processing are described below. The first

30   method combines color information before determining the edge pixels, while the second method determines edge pixels for each color plane and then combines the results into a set of composite edge pixels. The term "color plane" refers to an image, which is an N-dimensional array of pixel color values, where each pixel retains only one of a plurality

of the color image assigned to that pixel. For example, if each pixel were assigned a red value, a green value and a blue value, an image where each pixel only had its assigned red value is a color plane image for the red color plane.

In the first method, a composite gradient image comprising an array of gradient vectors, one per pixel, is computed from the color component values for the pixels. The composite gradient image is then processed to detect edge pixels.

One process for generating the composite gradient image generates a composite gradient for each pixel based on the gradients at that pixel in each of the color planes, where the composite gradient for a pixel is a modified vector addition of the gradients in each color plane at that pixel. The modified vector addition is modified in that the signs of the individual vectors are changed as needed to keep the directions of all of the addends within one half plane when there are more than two color planes.

Fig. 10 is used to illustrate such a modified vector addition. As shown there, the vectors in each color plane for pixel A have a range of directions that is less than one half circle, so the composite gradient is just the vector sum of the vectors in each of the components. The same is true of pixels B and C. However, there is no orientation of a half circle that would contain the directions of all of the vectors for pixel D. In the latter case, the sign of one of the vectors is reversed (i.e., pointed in the opposite direction), before the vectors are added.

In effect, this modified vector addition takes into account that for any given point on a given edge, there are two vectors that define the normal to the edge or the tangent to the edge. Consequently, a gradient vector can be reversed and still represent the same edge. By ensuring that the vectors being added all fall within a half circle, the contribution of one component gradient vector is less likely to cancel out the contribution of another component gradient vector.

By ensuring that the addends fall within a half circle, the polarities of the gradients should not greatly affect the outcome of their sum. In many images, the change of the polarity corresponds to a physical characteristic of the image. The polarity of edges in each color component of a YUV image are generally independent of each other. For example, if a bright green region transitions to a dark red region, "Y" decreases and "V" increases.

In some color spaces, the color components might have more influence in evenly weighted addition, due to overall color differences or due to a tendency, in some color spaces, to have more extreme gradient magnitudes, reflective of more pronounced or

stronger edges. In such cases, the color components might be weighted with a normalization factor before being added. Good normalization factors are scalar values that result in similar weights for gradient magnitudes in each color plane. For example, where the dynamic ranges between a luminance and chrominance color planes is such

5   that the average gradient magnitude is twice as much in the luminance color plane relative to a chrominance color plane over a sampling of images or a single image, the luminance vector might be normalized by dividing by two before adding the vectors.

Another process for generating the composite gradient image generates, for each pixel, a composite gradient that is equal to, or a function of, the color component vector at

10   that pixel with the greatest magnitude. In this process of taking the largest vector instead of adding vectors, a normalization factor might be applied before the comparison is done to select the vector with the largest magnitude.

However the composite gradient vectors are generated, they collectively form a composite gradient image. From that composite gradient image, edge pixels can be

15   determined and edge chains formed of those edge pixels in order to perform a segmentation process on the image.

4. Multiclass Edge Chaining

As explained above, once the edge pixels of an image are identified, those edge pixels are used to identify edge chains and those edge chains are used in a segmentation

20   process. In an edge chain identification ("edge chaining") process described herein, edge pixels are classed into a plurality of classes. In the example described here in detail, with reference to Fig. 11, the plurality of classes is two classes, designated "strong edge pixels" and "weak edge pixels". Edge pixels are the approximate points in an image where the color values undergo relatively large shifts, such as the shift in image 1110 from pixel

25   1102(1) to pixel 1104(1).

Two methods of identifying edge pixels of each class are described below: the gradient method and the Laplacian method. The gradient method is illustrated with reference to Figs. 12-13, while the Laplacian method is illustrated with reference to Fig. 14.

30   In the gradient method, midpixels that are local maxima of gradients of color values in the gradient direction are identified as edge pixels. A local maximum is a point where the value of a function is higher than the function value at surrounding points on

15

one side, and higher than or equal to the function value at surrounding points on the other side.

For example, consider Fig. 12, which shows an array of midpoints of an image -- the image's image pixels are not shown. Several midpoints are shown, some of which are labelled "A" through "I". In the gradient process for determining whether or not edge point A is an edge pixel and, if so, which class of edge pixel, the gradients of the midpixels A through I are considered.

In the gradient method, a processor selects a second midpixel from among midpixels B through I that is closest to a ray originating from midpixel A pointed in the same direction as the direction of the gradient of midpixel A. A tie breaking rule might be used to selection a closest midpixel if two midpixels are equidistant from the ray. In the example of Fig. 12, the closest midpixel is midpixel C. There is a third midpixel that is collinear with midpixel A and the closest midpixel C. In this example, that third midpixel is pixel H. A gradient can be defined at each midpixel, including midpixels A, C and H. Arrows are included in Fig. 12 to illustrate the magnitude and directions of the gradients for those three midpixels.

If the magnitude of midpixel A's gradient is larger than the magnitude of the gradient of one of midpixel C or midpixel H and larger than or equal to the gradient magnitude of the other one of midpixel C or midpixel H, then midpixel A is identified as a strong edge pixel.

If midpixel A is not identified as a strong edge pixel, it is tested to determine if it meets the criteria for a weak edge pixel. Fig. 13 illustrates a process for determining whether a midpixel is a weak edge pixel. As shown there, midpixel A is being considered for weak edge pixel status. To do this, consider a line passing through midpixel A and parallel to midpixel A's gradient direction, shown by line 1302 in Fig. 13. The neighboring midpixels B through I define a square and line 1302 intersects that square at two points, shown as points 1304(1) and 1304(2) in Fig. 13.

The gradients at midpixel A and each of points 1304(1) and 1304(2) can be found through interpolation. If midpixel A's gradient magnitude is greater than the gradient magnitudes of one of point 1304(1) or point 1304(2) and greater than or equal to the gradient of the other of point 1304(1) or point 1304(2), then midpixel A is designated a weak edge pixel. Otherwise, midpixel A is an undesignated midpixel.

In a typical image, more edge pixels are weak edge pixels than strong edge pixels, as weak edge pixels generally correspond to gradual changes in color or intensity in the

image or sharp contrasts, while strong edge pixels generally correspond usually only to sharp and unambiguous contrasts.

Referring now to Fig. 14, the Laplacian method will now be described with reference to determining the edge pixel status of a midpixel 1405 located within the rectangle defined by four image pixels 1410, 1420, 1430, 1440. Each of the four image pixels are labelled with a sign "+" or "-" representing the sign of the Laplacian function at that image pixel. The Laplacian is a second vector derivative. In an image, zero crossings of the Laplacian function applied to the pixel color values are edge pixels. The zero crossings of the Laplacian occur where the second derivative of the Laplacian is zero.

If the rectangle enclosing midpixel 1405 contains a zero crossing, midpixel 1405 is an edge pixel. With discrete pixel locations, calculating the location of a zero crossing is either not possible or is computationally difficult. However, the signs of the Laplacian at the image pixels are indicative of the likelihood of the zero crossing being in the rectangle. Consequently, the Laplacian method identifies a midpixel as a strong edge pixel if the signs of the Laplacian function at each of the four surrounding image pixels are different both vertically and horizontally (i.e., the upper right and lower left pixel have one sign and the lower right and upper left have another sign). If the signs are such that there is one instance of one sign and three instances of the other sign, or if there are two instances of each sign that are not both different vertically and horizontally, then the midpixel is identified a weak edge pixel. If the signs are the same for all four image pixels, the midpixel is identified as not being an edge pixel.

Two methods of identifying multiple classes of edge pixels have been described, but other suitable methods might be used instead. An example of a result of identifying classes of edge pixels is illustrated in Fig. 15. The image in Fig. 15 comprises two segments, an interior segment of black pixels and an exterior segment of white pixels. The results of an edge pixel detection process are shown, with identified strong edge pixels and identified weak edge pixels shown; nonedge midpixels are omitted from Fig. 15. In general, strong edge pixels are midpixels that correspond to a large and/or definite shifts in the image (e.g., sharp contrasts), while weak edge pixels reflect subtle changes in the image that may be due to actual color edges or might be caused by color bleeding or noise.

Once the edge pixels of differing class are identified, edge chains can be identified. An edge chain is identified as an ordered set of edge pixels. One way to

17

identify edge chains from edge pixels of differing classes is to start with an arbitrary strong edge pixel and add it to an edge chain as one end of the edge chain. Then look to the nearest neighboring midpixels of the added edge pixel for another strong edge pixel. If one exists, extend the edge chain to the neighbor, by adding the edge pixel to the

5   growing end of the edge chain. If more than one strong edge pixel neighbor exists, select the one with the closest gradient that is perpendicular to the direction of the gradient of the edge pixel. As a further test, the magnitude of the gradient at each neighbor can be considered. If no strong edge pixels exist among the nearest neighbors, extend the edge to a neighbor that is a weak edge pixel and if more than one exists, select the best choice

10   using a test similar to that for selecting among multiple strong edge pixels. If no strong or weak edge pixels exist to continue the edge chain, the chain stops and the process seeks another arbitrary strong edge pixel.

The process repeats until all the edge pixels have been processed or considered, resulting in a set of edge chains for an image. These edge chains can then be used in a

15   segmentation process to segment an image.

5. Multiscale Edge Chain Generation

With multiple transformed representations of an image, the coarser scale images can provide guidance to the finer scale images. The coarser scales can provide guidance when there is an association of edge pixels across the scales. Association of edge pixels

20   occurs when an edge pixel in one scale has the same or similar gradient magnitude and direction as the same edge pixel, or an adjacent edge pixel, in another scale. The sameness or similarity of two gradient vectors are tests that might take into account weighting factors between scales.

One multiscale edge chain generation process according to the present invention

25   uses edge pixels from a plurality of scale images to identify edge chains. Finer scale images tend to have more edges than coarser scale images, so a linking routine that identifies edge chains in a finer scale image will often encounter, at the end of an edge chain, multiple edge pixels to which the edge chain can extend. Some methods of deciding which edge pixel to select are described above. When multiple scale images are

30   available, the selection process can be refined.

In one approach, where a process is faced with a choice among the neighboring edge pixels and edge chains have been identified for coarser scales, the process favors in its selection edge pixels that would result in extending the edge chain along an edge chain

18

at coarser scales. For example, consider Fig. 16, which shows a midpixel array 1600, with edge pixels shown as filled circles and nonedge pixel midpixels as open circles. Suppose an edge chain has been identified and the end of the edge chain is at edge pixel 1605. The linking process is faced with a decision to extend the edge chain to edge pixel

5   1610 or edge pixel 1615. Suppose further that midpixel array 1600 is the result of edge pixel identification at a finer scale than a prior array and that in the prior, coarser array, an edge chain was identified that included edge pixels associated with edge pixel 1605 and edge pixel 1610, but not edge pixel 1615. In that case, the linking process would favor edge pixel 1610.

10   In some embodiments, the edge chains at coarser scales are determinative and edge pixel 1610 would be added to the edge chain without further inquiry. In other embodiments, other factors are taken into account along with the coarser scale correspondences. In yet other embodiments, a midpixel might be selected even if it is not an edge pixel, if an edge chain at a coarser scale indicates that the edge chain should be

15   continued over a gap, as illustrated in Figs. 17(a)-(b). As shown there, midpixel 1705 is not an edge pixel at the finer scale, so the edge chains 1710, 1712 would not reach midpixel 1705. However, at the coarser scale, the corresponding midpixel is an edge pixel and is included in an edge chain. Consequently, midpixel 1705 would be added to an edge chain at the finer scale, resulting in a continuation that joins edge chain 1710 and

20   edge chain 1715. Additionally, edge pixels from different finer scale edge chains may be associated with the same edge chain in a coarser scale. In such cases, two finer scale edge chains are joined in such a way as to duplicate as much of the coarse edge chain geometry as possible.

Further, as shown in Figs. 18(a)-(c), where an edge chain at a coarser scale is

25   longer than the associated chain in the finer scales, the linking process will lengthen the finer scale edge chain appropriately. This is referred to as "edge chain lengthening". In the example shown, two chains have the same edge pixels in the same order except that in the finer scale (Fig. 18(b)), the chain stops at midpixel 2302, while the coarser scale image edge chain continues one more midpixel from midpixel 2202 to midpixel 2201

30   (Fig. 18(a)). The process will then continue the finer edge chain to link to midpixel 2301 as shown in Fig. 18(c).

As illustrated in Figs. 19(a)-(b), once an edge chain has been completely generated, the linking routine might add a few more midpixels to each end of the edge. This often has the effect of closing small gaps between edge chains, and often completely

enclosing a given region. By adding one more pixel to each edge chain shown in Fig. 19(a), unenclosed regions become completely bounded, as shown in Fig. 19(b).

In a variation of the basic process for using multiscale associations in edge chain linking, multiple classes of edge pixels might be used. In one embodiment, edge chains are extended from an edge pixel end to another midpixel based on the class of the neighbors and any associations from coarser scales. For example, a linking process might first consider only strong edge pixels and look for a corresponding edge chain extension in a coarser scale, then considering weak edge pixels if not strong edge pixels or associations are found. In another variation, the linking process might first consider strong edge pixels then weak edge pixels and then only look for a corresponding edge chain extension in a coarser scale that uses strong or weak edge pixels, if no edge pixels are found at the finer scale.

6. Edge Chain Filtering

Fig. 20(a) is an edge chain image resulting from a process that identifies edge chains. In that edge chain image, there are many short edge chains that do not correspond to segment edges. Such spurious edges might occur when an edge chain is created primarily as a result of digitization errors or subtle changes in shading in the image.

Figs. 20(b)-(e) are the results after applying a threshold routine to the image at each scale. The threshold routine can generate the images of Figs. 20(b)-(e) using a static sliding scale or a dynamic sliding scale. In the static method, edge chains are removed if they do not meet both a baseline length requirement and a minimum intensity requirement.

The dynamic sliding scale test is more inclusive. The length and intensity thresholds are related. Thus, the longer a chain is, the lower the intensity threshold that must be met for retention of the edge chain. Similarly, the brighter the edge chain is, the lower the length threshold becomes.

In one implementation, edge chains are retained across scales despite thresholding. Specifically, edge chains are associated across scales and edge chains that survive thresholding in coarser scales are retained at finer scales, even if they would not survive thresholding at the finer scale if considered apart from the coarser scales. In another implementation, chains from previous video frames can prevent edge chains from being discarded.

20

Fig. 21 illustrates the situation where the origin of the input image is a video sequence. Note that the edge chain information at each scale is retained across the frames. Fig. 21(a) shows a prior frame of video; Fig. 21(b) shows a current frame of video; and Fig. 21(c) shows the current frame after thresholding based on the prior frame.

5   The prior frame is shown after thresholding, and edge chain 40011 was retained. The edge chains of the current frame include edge chains 40020, 40021, and 40022. Edge chains 40011 and 40021 are associated across frames because edge chains 40011 and 40021 share pixels with identical locations and similar gradients. In the current frame, edge chain 40020 passes the thresholding tests described above, so it will be retained.

10   However, neither edge chain 40021 nor 40022 pass the thresholding tests, so they are candidates for removal. To ensure consistent segmentation across frames, edge chain 40021 is retained since it is associated with edge chain 40011, which was retained in the previous frame. Conversely, edge chain 40022 is not retained because it failed the threshold tests and is not associated with an edge chain retained in the prior frame.

15   7. Combinations

          Six methods and corresponding apparatus for improving segmentation are described above. One or more of these methods and apparatus can be combined for greater improvement in the segmentation process. For example, progressive filling can be used over multiple scales, where the filling process is performed at the coarsest scale,

20   then the segments at the coarser scales are used to guide segmentation at the finer scales.

          Another combination is the combination of composite edge detection with multiple classes of edge detection. With that combination, an edge detection process would operate separately on each color plane to identify strong and weak edge pixels, then filter by combining edge pixels from different color planes.

25   In another variation, the color information is used at a later stage, after the strong and weak edge pixels have been identified, but before the edge chains are identified. An edge pixel identification routine creates two composite edge pixel images at each scale. The first image is of all the strong edge pixels from all color planes and the second image is of all the weak edge pixels from any color plane. Sometimes, the same edge pixel will

30   be designated in multiple color components. In that situation, the edge pixels are only included in the composite image once. Edge pixels in multiple color components are designated identical if one of two conditions is met, 1) they share identical location in

21

different color components, or 2) they share the same array locations in different color components and have similar gradients.

Yet another combination is a method wherein the edge chains are found in each component and then the edge chains are composited. In particular, the local extrema found in each color component are initially kept separate from the extrema found in the other color components of the image. The linking process creates edge chains by linking edge pixels in each color component separately, then combining the edge chains into one composite image. Because it is possible, even likely, that some of the edge chains determined by the gradients of one color component will be similar to edge chains determined by the gradients of the other color components, the linking process coalesces similar chains into one chain possibly forming longer chains with fewer gaps. Edge chains in the same scale are similar if they satisfy one or more of the following criteria: 1) identical in all respects; 2) share the majority of their pixels with each other; 3) identical in geometry (to within a small variance) but offset by very few pixels; 4) are both associated with the same coarser scale edge chain in another scale.

In yet another combination, all of the above processes are combined to form an improved segmentation process.

The above description is illustrative and not restrictive. Many variations of the invention will become apparent to those of skill in the art upon review of this disclosure. The scope of the invention should, therefore, be determined not with reference to the above description, but instead should be determined with reference to the appended claims along with their full scope of equivalents.

WHAT IS CLAIMED IS:

1   1. A method of detecting edges in a digital image, where the digital image

2   comprises an array of pixels each having a pixel location and a pixel color value and

3   where an edge represents a transition in the array between the pixels representing one

4   segment of the digital image and pixels representing another segment in the digital image,

5   the method comprising the steps of:

6   generating one or more filtered image, wherein each filtered image is a result of

7   filtering the digital image using a different filter for each of the plurality of

8   filtered images where the different filters provide different resolutions of

9   segments from the digital image;

10   identifying edges in the digital image by identifying color shifts among the array of

11   pixels in the digital image; and

12   identifying further edges in the digital image by identifying color shifts among the

13   pixels of the one or more filtered image.

1   2. The method of claim 1 wherein the one or more filtered image is a

2   plurality of filtered images including one filtered image generated using an identity filter,

3   thereby including the digital image as one of the plurality of filtered images.

1   3. The method of claim 1 wherein the different filters are filters of

2   different resolutions of scale.

1   4. The method of claim 1 wherein the different filters are smoothing filters

2   with different window sizes.

1   5. The method of claim 1 wherein the different filters decimate to

2   different pixel resolutions, resulting in varying pixel array sizes over the one or more

3   filtered image.

1   6. The method of claim 1 wherein the filters are gradient filters of various

2   polynomial degrees of gradients.

1   7. The method of claim 1 wherein the different filters are Laplacian filters.

1   8. The method of claim 1 wherein the digital image is represented as a

2   single color plane and each pixel color value is represented by a scalar value representing

3   a shade in the single color plane.

1   9. The method of claim 8 wherein each pixel color value for a pixel is a

2   multicomponent vector and the scalar value for the pixel is a function of the components

3   of the pixel color value vector.

1   10. The method of claim 9 wherein the function of the components of the

2   pixel color value vector is a luminance function.

1   11. The method of claim 9 wherein the function of the components of the

2   pixel color value vector is a sum.

1   12. The method of claim 9 wherein the function of the components of the

2   pixel color value vector is a weighted sum.

1   13. A method of detecting edges in a digital image, where the digital

2   image comprises an array of pixels each having a pixel location and a pixel color value

3   and where an edge represents a transition in the array between the pixels representing one

4   segment of the digital image and pixels representing another segment of the digital image,

5   the method comprising the steps of:

6   identifying edge points of the digital image, where an edge point is a point in the

7   array of pixels between point locations that are pixel locations;

8   identifying a plurality of classes of edge points, including a more certain class of

9   edge points and a less certain class of edge points where an edge point in the

10   more certain class is more likely to be representative of an edge of a segment of

11   the digital image than an edge point in the less certain class; and

12   identifying edges by linking adjacent edge points of the more certain class for

13   forming edge point chains;

14   when gaps are present between edge point chains, identifying edge points in the less

15   certain class to span gaps in the edge point chains; and

16   using the edge point chains as representations of the edges in the digital image.

1   14. The method of claim 13 where the pixel array is a rectangular array,

2   further comprising a step of locating each edge point at the center of a rectangle defined

3   by four mutually adjacent pixels in the pixel array.

1   15. The method of claim 13 where the plurality of classes include more

2   than two classes and each class has an relative certainty associated with the edge points in

3   the class with the relative certainties of each class being orderable and distinct from the

4   relative certainties of other classes, the method further comprising a step of identifying

5   edge points in a class for use in joining edge chains, before edge points in less certain

6   classes are used, but after edge points in more certain classes are used.

1   16. A method of detecting edges in a digital image, where the digital

2   image comprises an array of pixels each having a pixel location and a pixel color value,

3   where a pixel color value is representable as a vector in a color space, and where an edge

4   represents a transition in the array between the pixels representing one segment of the

5   digital image and pixels representing another segment of the digital image, the method

6   comprising the steps of:

7   separating the digital image into a plurality of color plane images;

8   filtering each of the color plane images to form an extrema image for each color

9   plane, an extrema image being an array of values corresponding to the array of

10   pixels where a value in the array of values represents an extrema value at a

11   corresponding location in array of pixels;

12   combining the plurality of extrema images to forms a composite extrema image; and

13   using the composite extrema image to detect edges in the digital image.

1   17. The method of claim 16 wherein the plurality of color planes is two or

2   more color planes.

1   18. The method of claim 16 wherein the plurality of color planes is three

2   color planes.

1   19. The method of claim 16 wherein the plurality of color planes

2   comprises a luminance color plane and two chrominance color planes.

1   20. The method of claim 16 wherein the plurality of color planes
2   comprises a red color plane, a blue color plane and a green color plane.

1   21. The method of claim 16 wherein the plurality of color planes
2   comprises a cyan color plane, a yellow color plane, a magenta color plane and a black
3   color plane.

1   22. A method of detecting edges in a digital image, where the digital
2   image comprises an array of pixels each having a pixel location and a pixel color value,
3   where each pixel color value is representable as a vector in a color space, and where an
4   edge represents a transition in the array between the pixels representing one segment of
5   the digital image and pixels representing another segment of the digital image, the method
6   comprising the steps of:
7   separating the digital image into a plurality of color plane images, wherein a color
8   plane image is an array of pixels with pixel values corresponding to one
9   component of a pixel color value vector;
10   generating one or more filtered image, wherein each filtered image is a result of
11   filtering one of the color plane images using a different filter for each of the
12   plurality of filtered images where the different filters provide different
13   resolutions of segments from the digital image;
14   identifying edges in the digital image by identifying color shifts among the array of
15   pixels in the color plane images; and
16   identifying further edges in the digital image by identifying color shifts among the
17   pixels of the one or more filtered image.

1   23. A method of detecting edges in a digital image, where the digital
2   image comprises an array of pixels each having a pixel location and a pixel color value,
3   where each pixel color value is representable as a vector in a color space, and where an
4   edge represents a transition in the array between the pixels representing one segment of
5   the digital image and pixels representing another segment of the digital image, the method
6   comprising the steps of:
7   identifying edge points of the digital image, where an edge point is a point in the
8   array of pixels between point locations that are pixel locations;

26

9   identifying a plurality of classes of edge points, including a more certain class of

10   edge points and a less certain class of edge points where an edge point in the

11   more certain class is more likely to be representative of an edge of a segment of

12   the digital image than an edge point in the less certain class, wherein the class of

13   an edge point is a function of the component values of pixel color value vectors

14   for pixels near the edge point; and

15   identifying edges by linking adjacent edge points of the more certain class for

16   forming edge point chains;

17   when gaps are present between edge point chains, identifying edge points in the less

18   certain class to span gaps in the edge point chains; and

19   using the edge point chains as representations of the edges in the digital image.

1   24. A method of segmenting a digital image into a plurality of segments

2   defined by edges of objects in the digital image, where the digital image comprises an

3   array of pixels each having a pixel location and a pixel color value and where an edge

4   represents a transition in the array between the pixels representing one segment of the

5   digital image and pixels representing another segment of the digital image, the method

6   comprising the steps of:

7   identifying a set of edge chains, where an edge chain represents a border between

8   two segments and at least one edge chain is not part of a closed loop enclosing

9   and defining a segment;

10   setting a maximum allowable threshold for gaps in edge chains;

11   positioning a first brush in a first location in the array such that the first brush does

12   not overlap any edge chains, wherein a brush is a pixel window movable over

13   the array and wherein the first brush is a pixel window of a size such that the

14   first brush cannot pass through a gap in edge chains that no wider than the

15   maximum allowable threshold without the first brush overlapping an edge chain;

16   associating pixels with a segment where each pixel associated with the segment is a

17   pixel reachable by the first brush from the first location without the first brush

18   having to pass through a gap smaller than a width of the first brush;

19   associating other pixels with other segments using the first brush positioned in

20   second and subsequent locations in the array; and

21   repeating the steps of positioning and associating using a second brush that is smaller

22   than the first brush, to associate pixels that were not already associated in the

23   steps of positioning and associating.

1   25. The method of claim 24, wherein the first brush is a square pixel

2   window with each side of the square being a number of pixels one greater than the

3   maximum allowable threshold.

1   26. The method of claim 24, wherein the step of repeating is performed

2   with successively smaller brushes until a brush with an area of a single pixel is used.

1   27. The method of claim 26, further comprising a step of assigning

2   unassigned pixels to segments when unassigned pixels remain after the processes of

3   repeating have been performed for each brush.
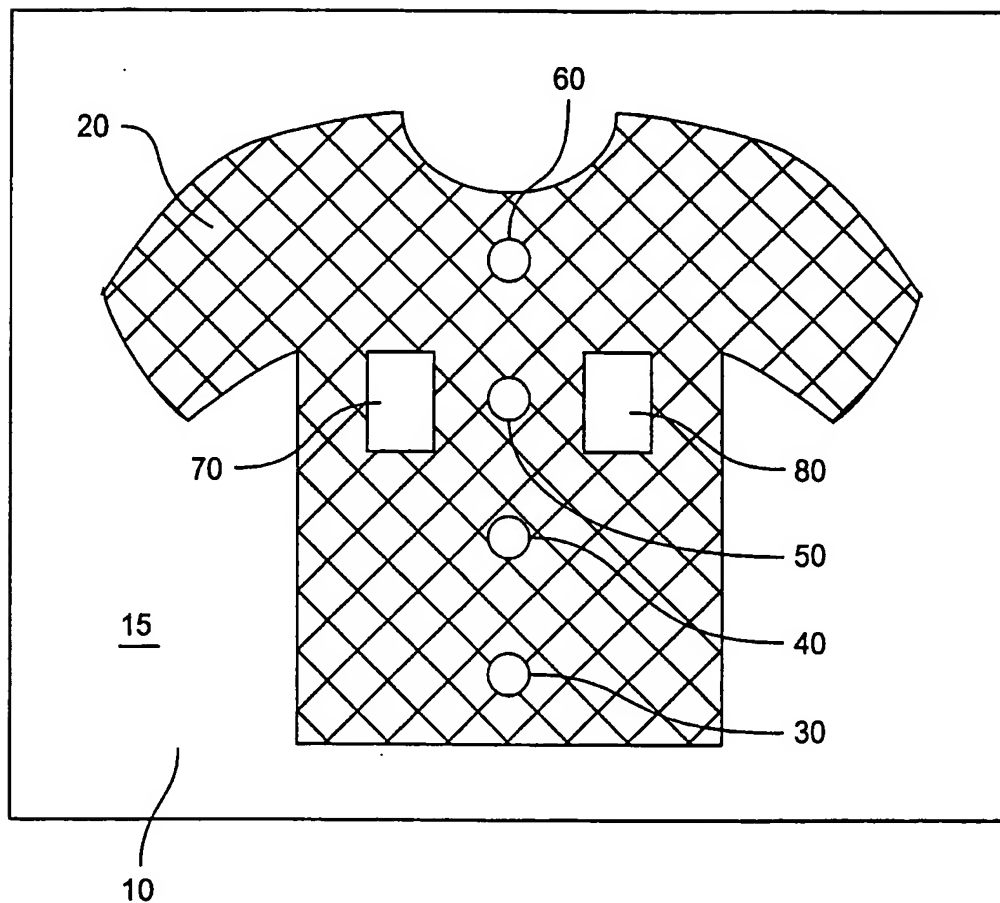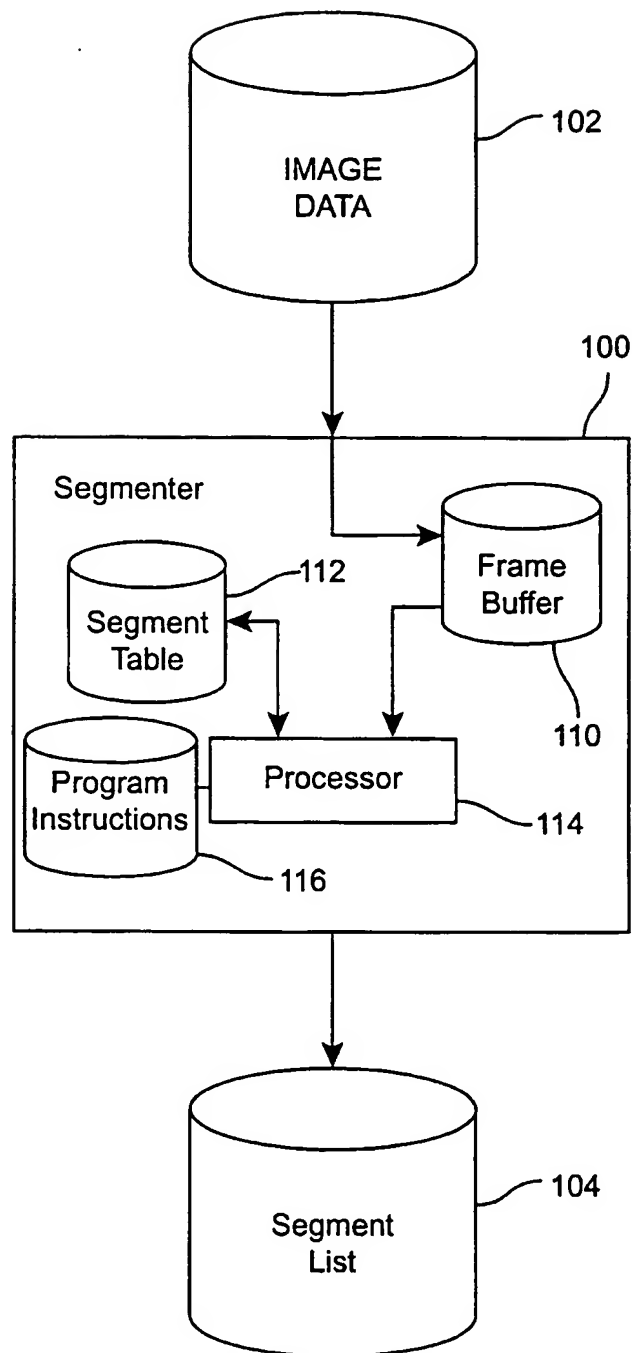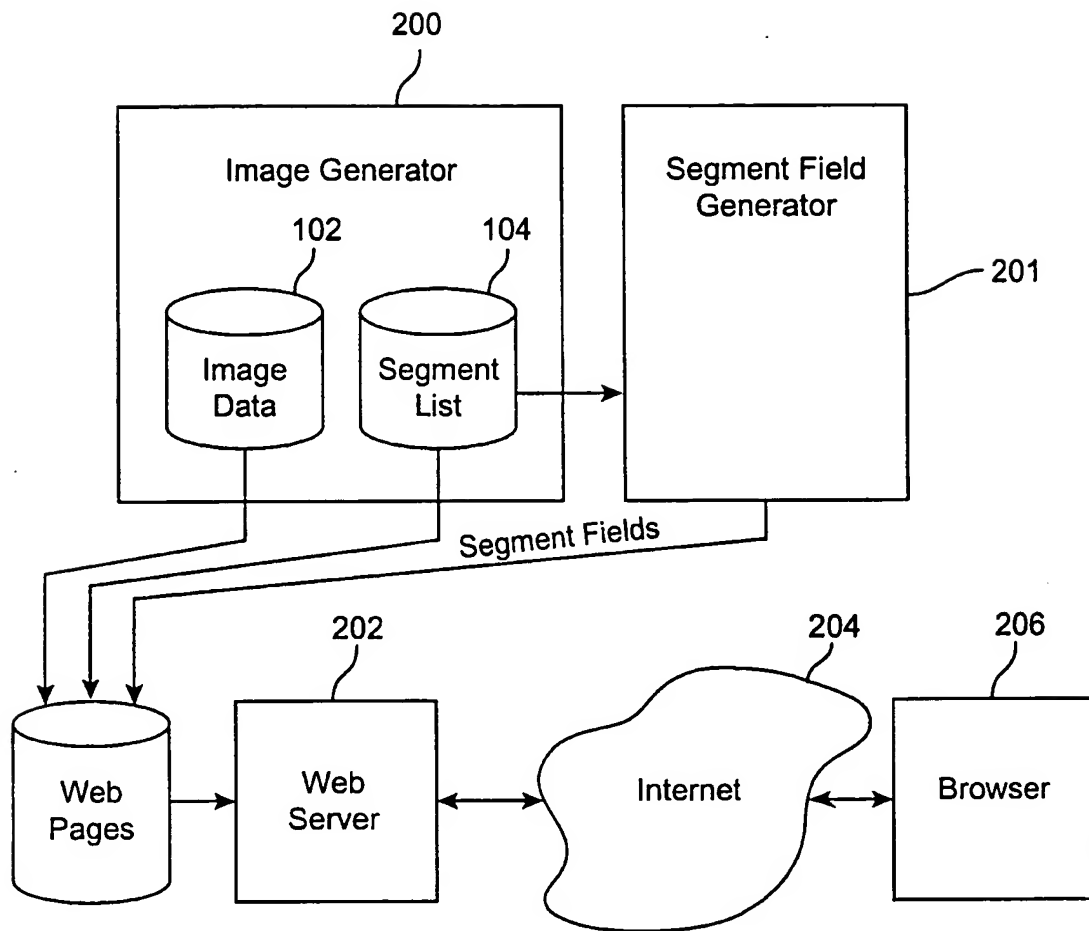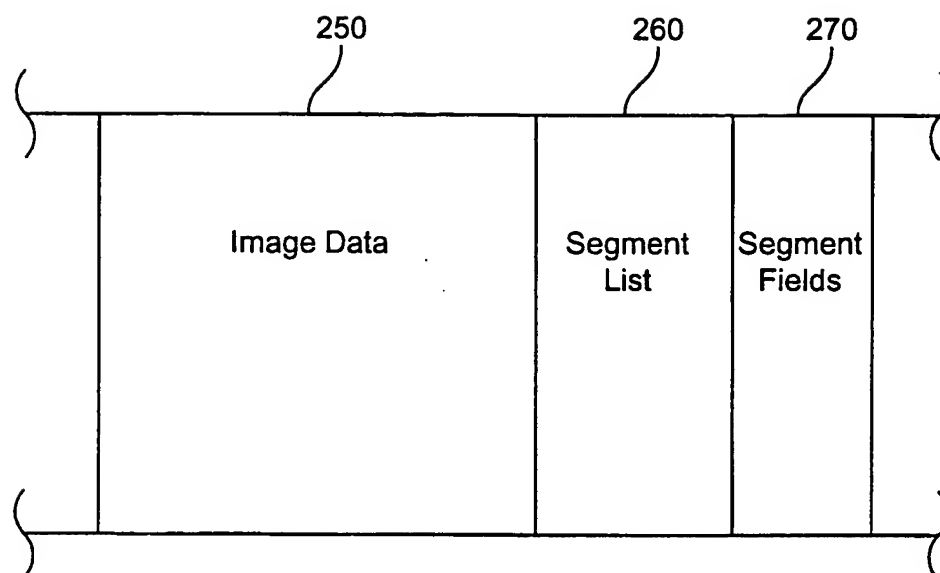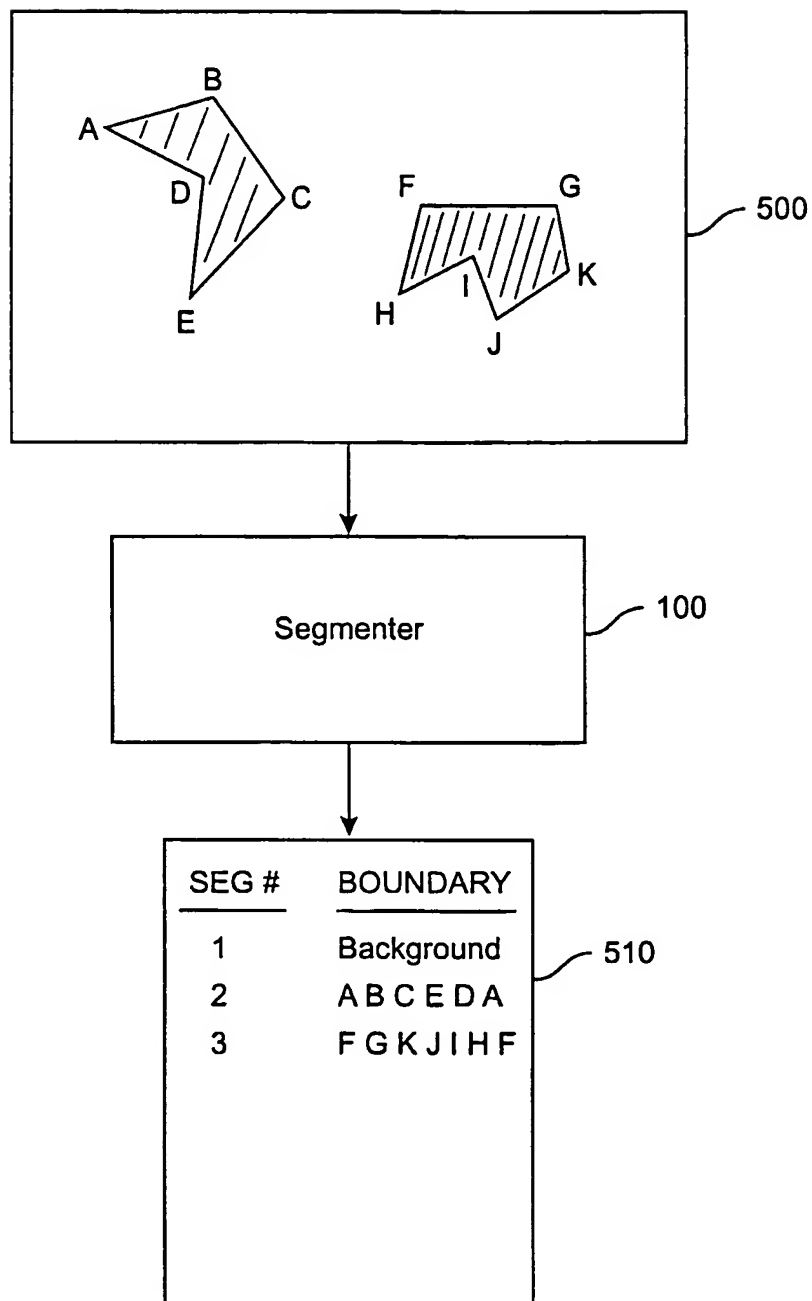
FIG. 1

FIG. 2

FIG. 3

250   260   270

Image Data	Segment List	Segment Fields

FIG. 4

FIG. 5

FIG. 6

FIG. 7

FIG. 9

Brush

Image for Segmentation
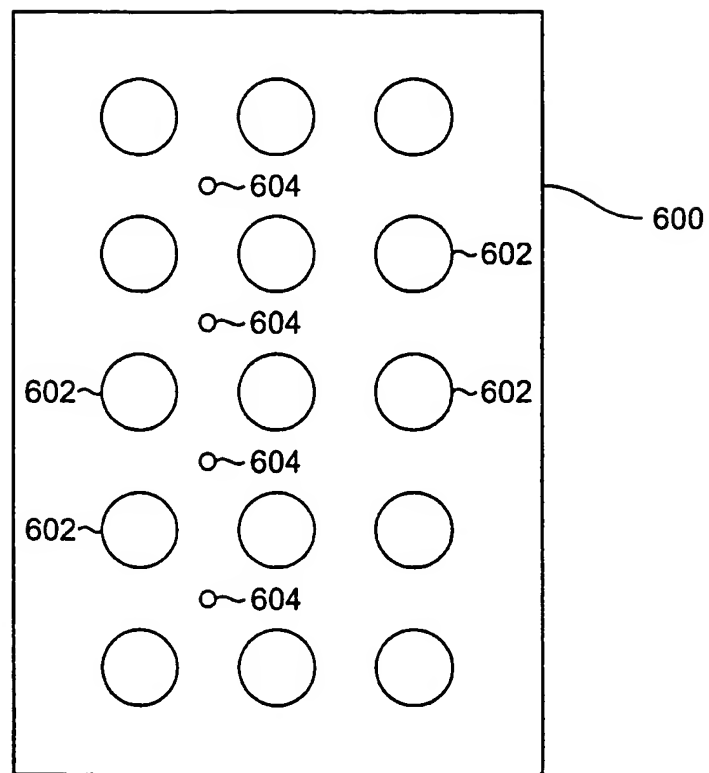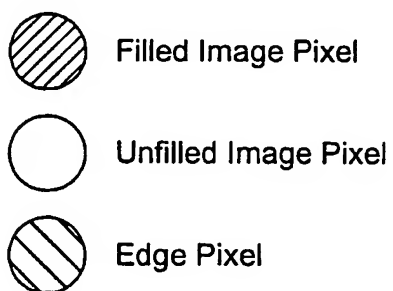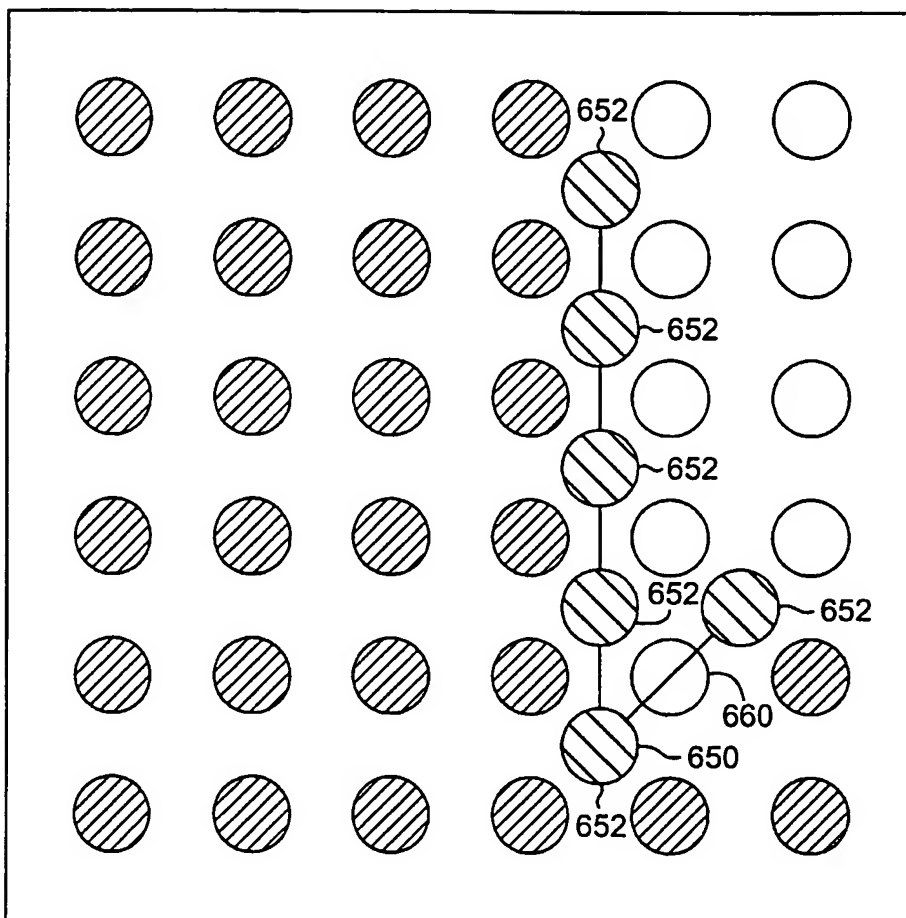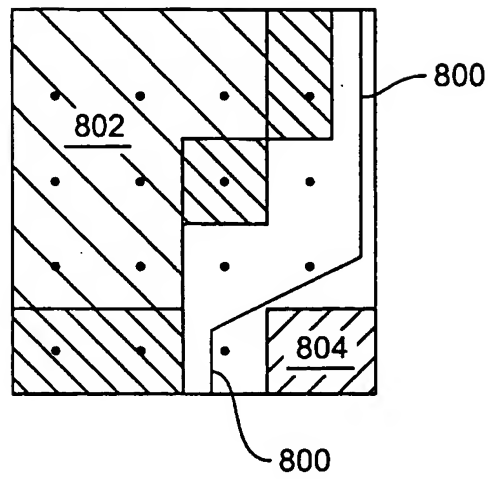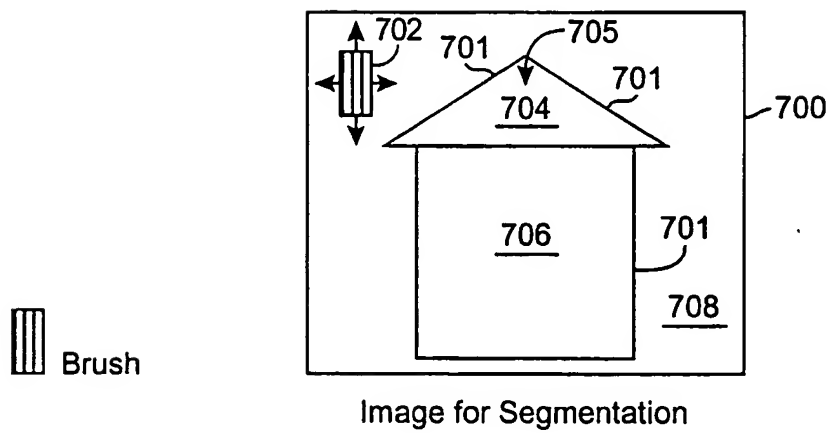
FIG. 8
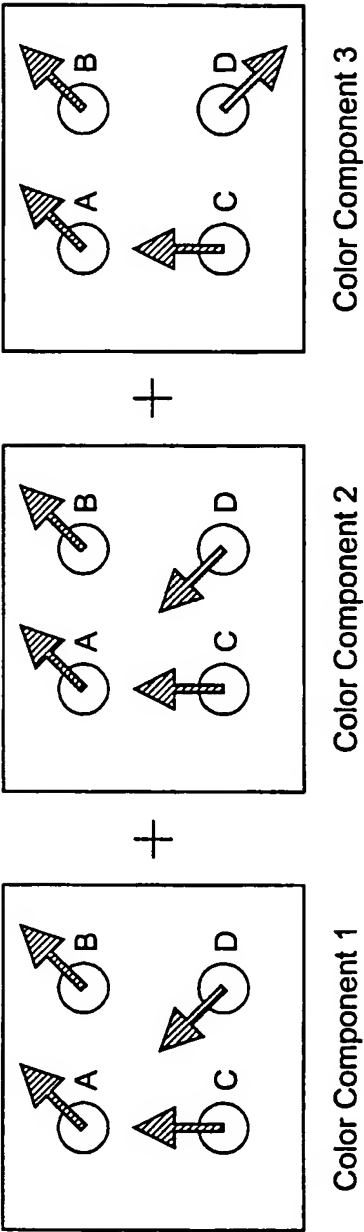
FIG. 10
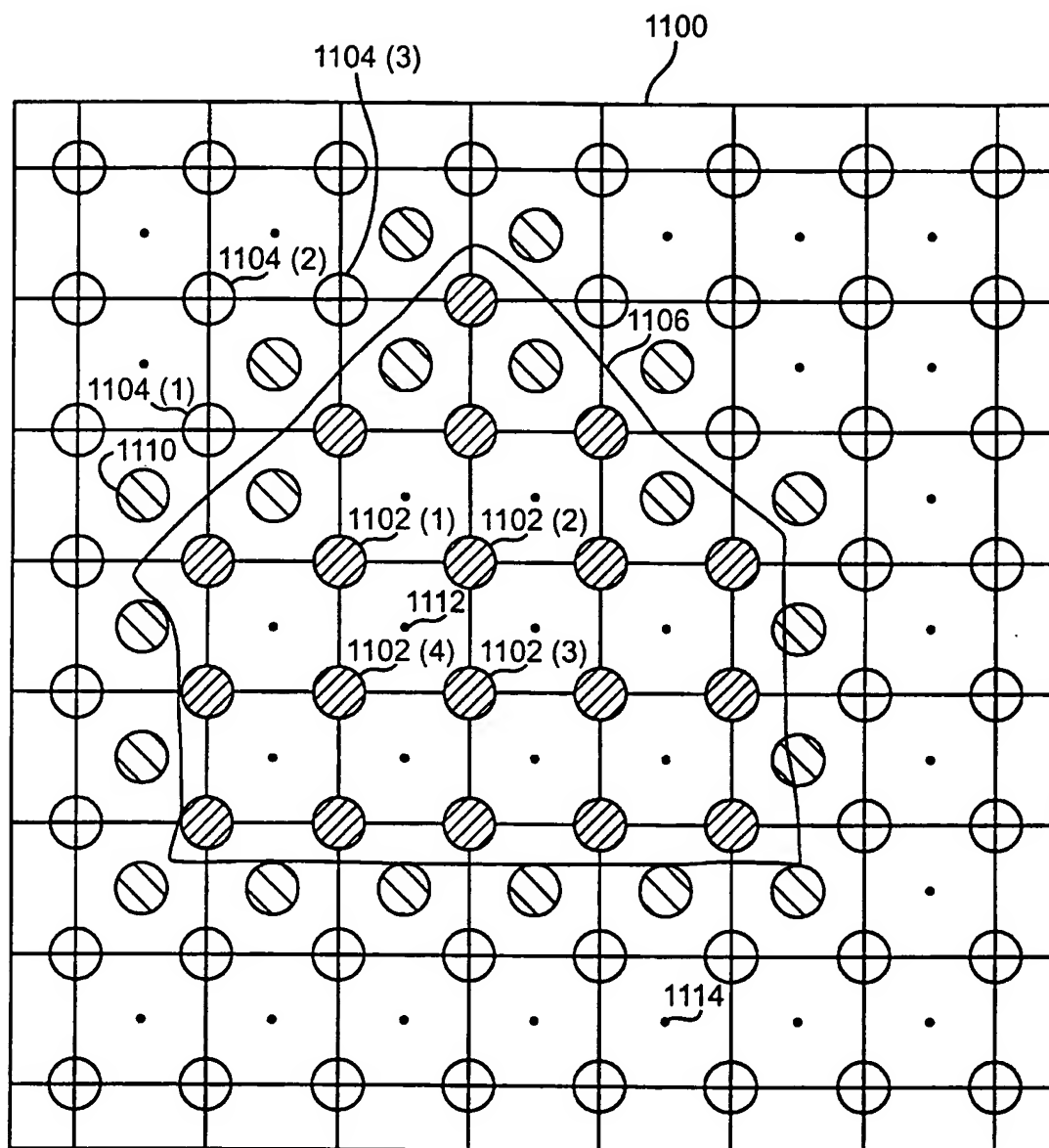
FIG. 11
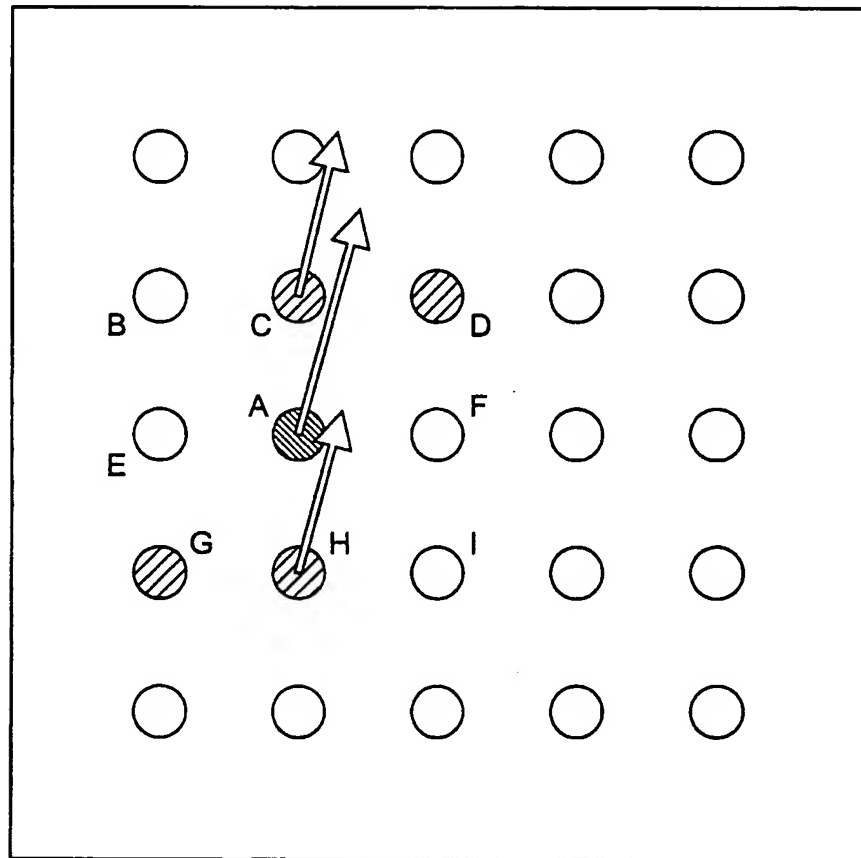
FIG. 12
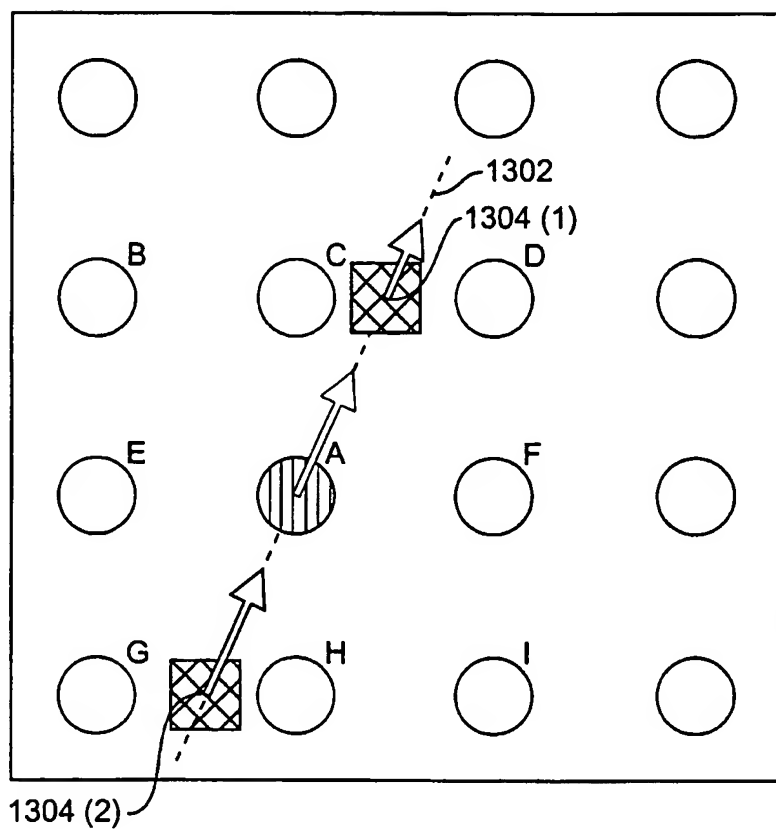
FIG. 13
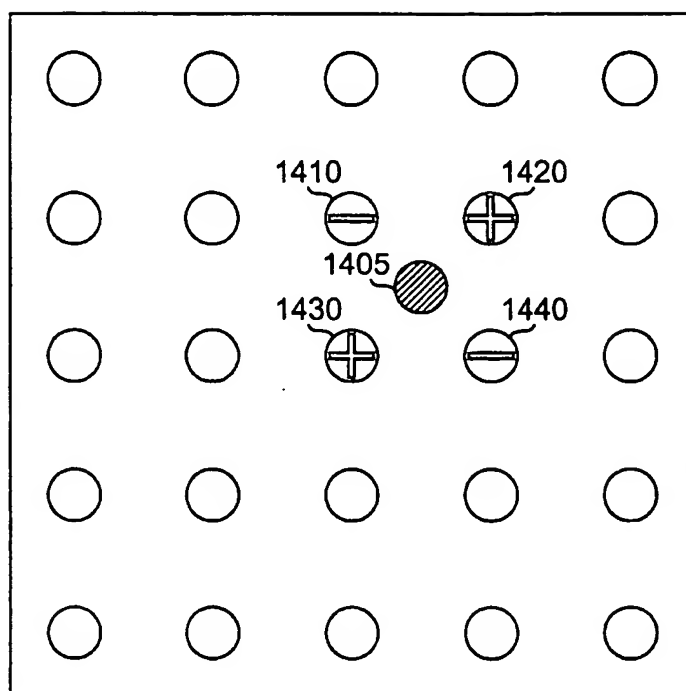
FIG. 14
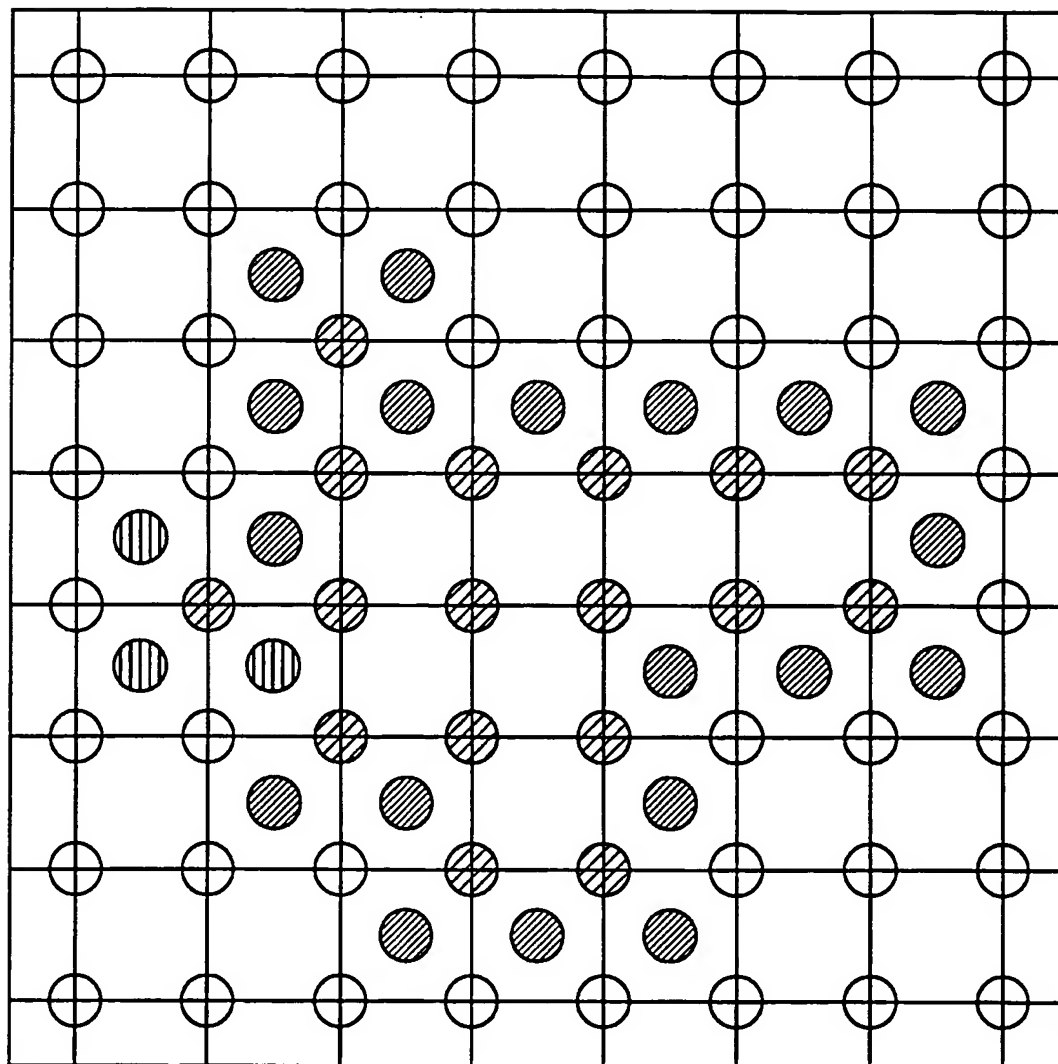
Strong Edge Pixel

Weak Edge Pixel

Interior Image Pixel

Background Image Pixel

FIG. 15

FIG. 16

Scale 2 Coarser scale

FIG. 17(b)

Scale 1 Finer scale
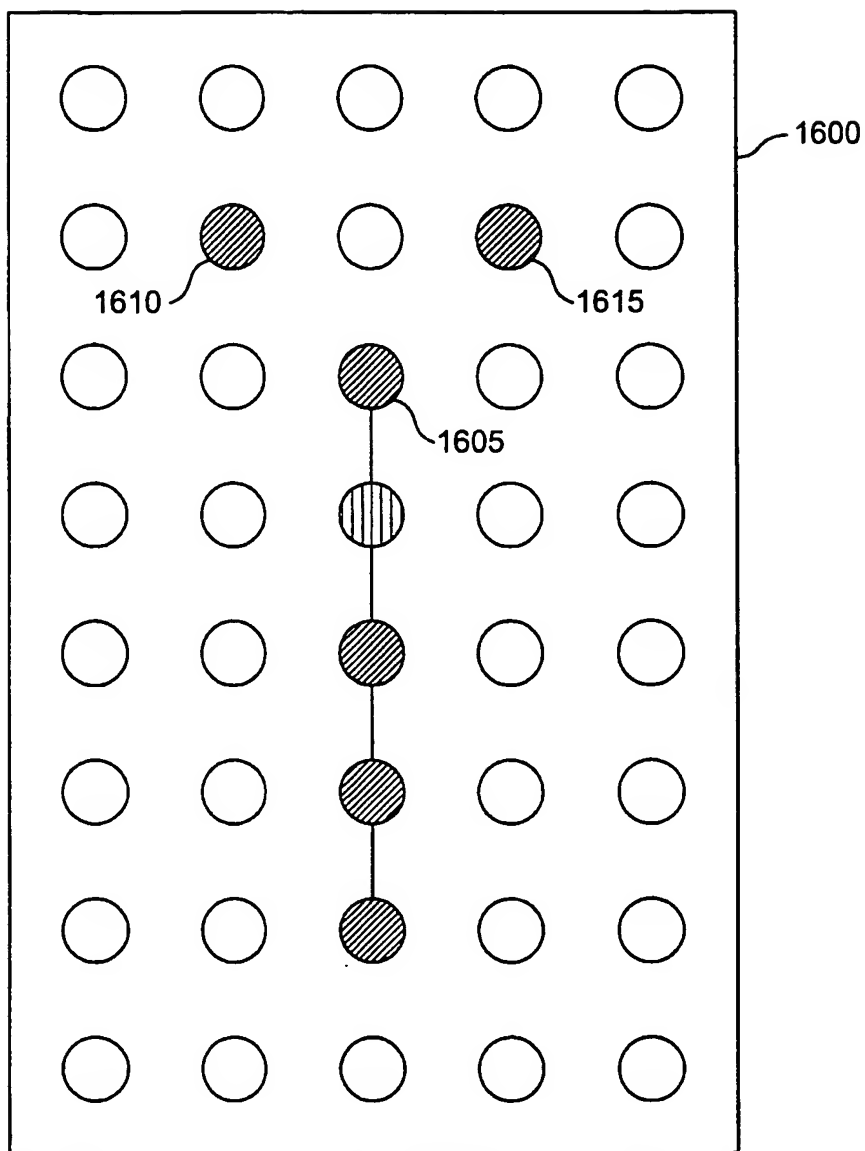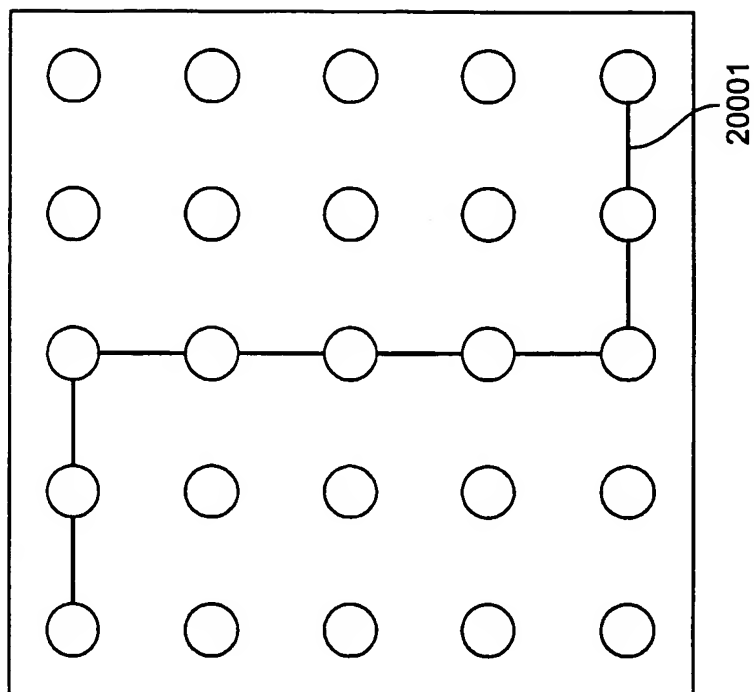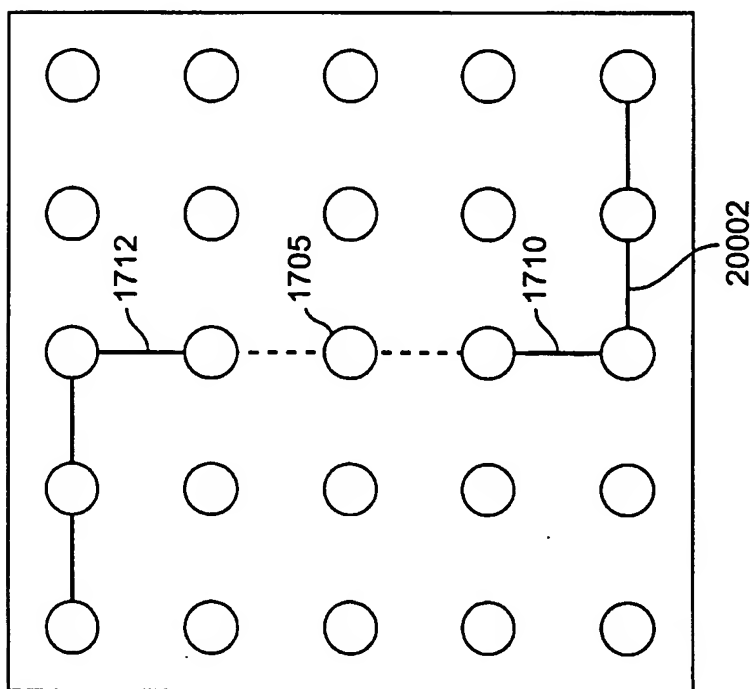
FIG. 17(a)
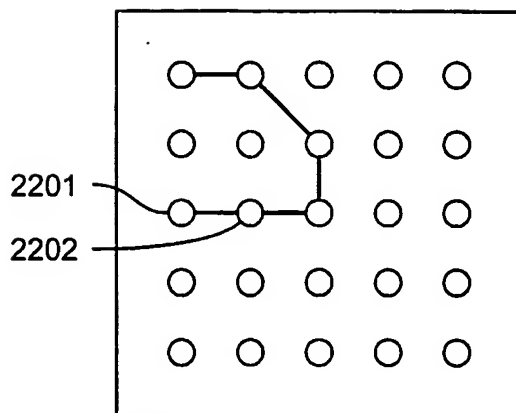
- - - - - Chain

FIG. 18(a)

Scale 2 (Coarser)
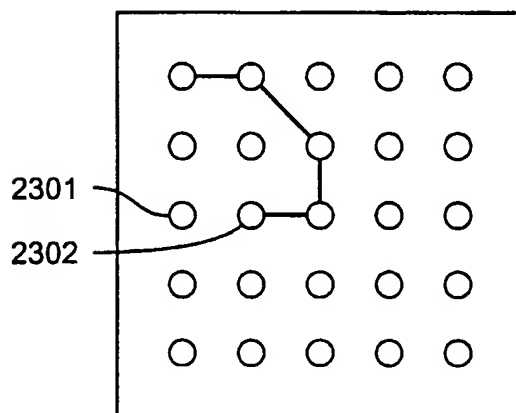
FIG. 18(b)

Scale 1
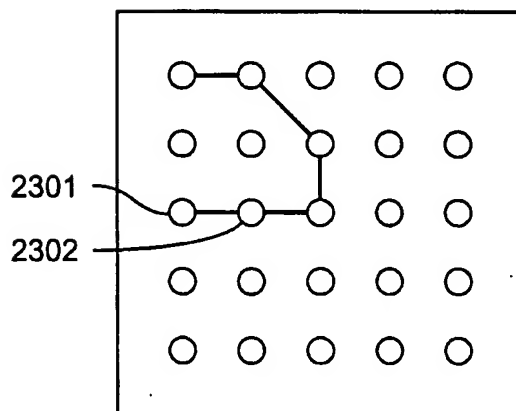
FIG. 18(c)

Scale 1 - after extension of
edge chain based on scale 2
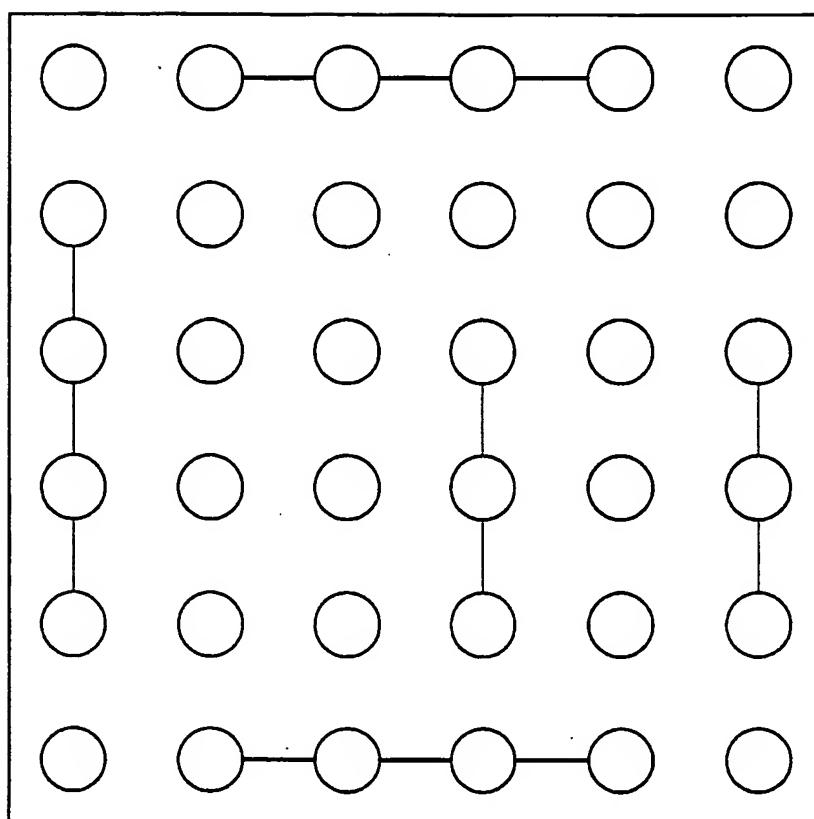
FIG. 19(a)
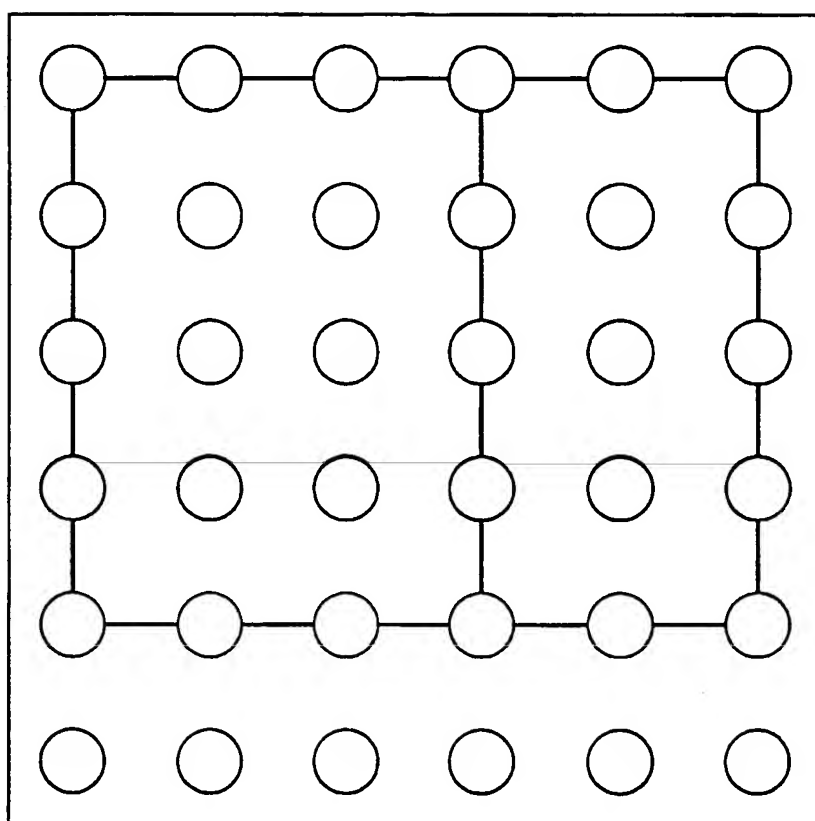
FIG. 19(b)

FIG. 20a

21 / 25

FIG. 20b
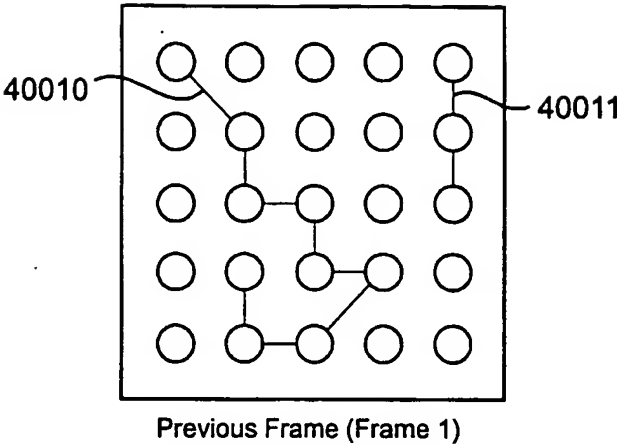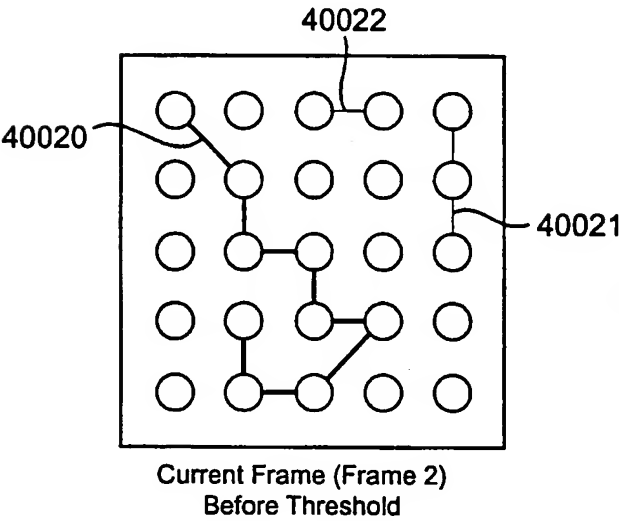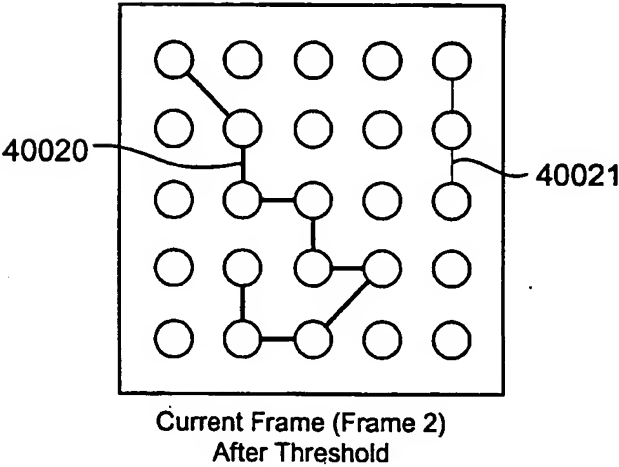
FIG. 20c

FIG. 20d

FIG. 20e

FIG. 21(a)

Previous Frame (Frame 1)

FIG. 21(b)

Current Frame (Frame 2)
Before Threshold

FIG. 21(c)

Current Frame (Frame 2)
After Threshold

INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7   G06T5/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7   G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

WPI Data, PAJ, INSPEC, EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	COOK G W ET AL: "MULTIRESOLUTION SEQUENTIAL EDGE LINKING" PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING. (ICIP),US,LOS ALAMITOS, IEEE COMP. SOC. PRESS, 23 October 1995 (1995-10-23), pages 41-44, XP000624176 ISBN: 0-7803-3122-2 abstract page 41, paragraph 1 page 42, paragraph 2.1 page 42, paragraph 2.2 ‑‑‑ -/--	1-27

[X]   Further documents are listed in the continuation of box C.   [X]   Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
27 October 2000	06/11/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340–2040, Tx. 31 651 epo nl, Fax: (+31-70) 340–3016	Gonzalez Ordonez, O

Form PCT/ISA/210 (second sheet) (July 1992)

1

INTERNATIONAL SEARCH REPORT

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	LI M ET AL: "PYRAMID EDGE DETECTION FOR COLOR IMAGES" OPTICAL ENGINEERING,US,SOC. OF PHOTO-OPTICAL INSTRUMENTATION ENGINEERS. BELLINGHAM, vol. 36, no. 5, 1 May 1997 (1997-05-01), pages 1431-1437, XP000692376 ISSN: 0091-3286 abstract page 1431, line 9 - line 21 page 1434, paragraph 4 ---	1-27
A	EP 0 853 293 A (CANON KK) 15 July 1998 (1998-07-15) abstract column 7, line 5 - line 9 column 8, line 4 - line 21 column 12, line 16 -column 13, line 3 -----	1-27

INTERNATIONAL SEARCH REPORT

		Internal Application No
		PCT/US 00/15942

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0853293 A	15-07-1998	JP 10191020 A	21-07-1998